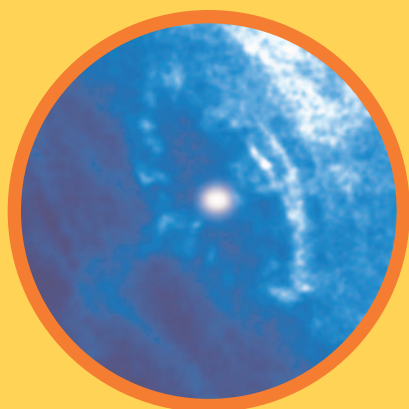
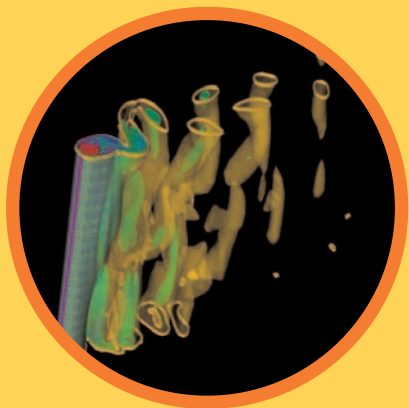


COMPOSE

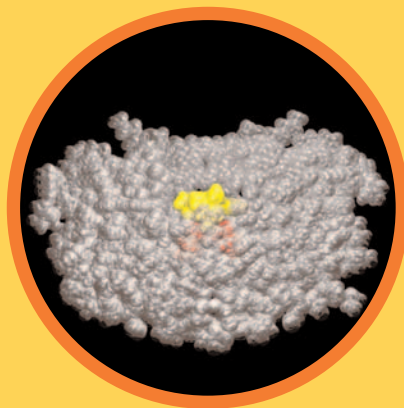
THE DOE CSGF ANNUAL ESSAY CONTEST JOURNAL



NOT ENOUGH MILK IN THE MILKY WAY



SHADES OF RIGHT



THE DREAM OF THE DRUG-DESIGNING MACHINE

The DOE CSGF Annual Essay Contest was launched in 2005 as an exciting opportunity for DOE CSGF Fellows to hone their writing skills. This contest requires Fellows to write a popular science essay on a topic of personal importance written for a non-science audience.

The DOE CSGF is proud to recognize outstanding Computational Science Graduate Fellows who have completed a non-technical writing composition on a topic in computational science ranging from 700-1000 words. In addition to recognition and a cash prize, the winners received the opportunity to work with a professional science writer to critique and copy-edit their essays.

These copy-edited winning essays are published here, in this inaugural issue of [Compose Magazine](#).

FOR MORE INFORMATION ON THE DOE CSGF ANNUAL ESSAY CONTEST, VISIT
WWW.KRELLINST.ORG/CSGF/ESSAY2005.SHTML



Award winner Ian Parrish with Walt Polansky, DOE Office of Advanced Scientific Computing Research. Ian received his award and \$1000 cash prize during an awards luncheon in Seattle, Washington on November 14, 2005.

Twenty essay submissions were received in 2005, and three were selected as finalists. Ian Parrish was selected as the winning author for his essay, “Not Enough Milk in the Milky Way.” Honorable Mention was awarded to Mala Radhakrishnan for “The Dream of the Drug-Designing Machine” and Kristine Cochran for “Shades of Right.”

Page 3 – Not Enough Milk in the Milky Way

By Ian Parrish, a second-year fellow studying Computational Plasma Physics at Princeton University.

Page 5 – The Dream of the Drug-Designing Machine

By Mala Radhakrishnan, a second-year fellow studying Physical Chemistry at the Massachusetts Institute of Technology.

Page 6 – Shades of Right

By Kristine Cochran, a fourth-year fellow studying Structures at the University of Illinois – Urbana/Champaign.

**Ian Parrish – 2005
DOE CSGF
Essay Contest
Winner**



Not Enough Milk in the Milky Way

As I crawled out of my tent in the desert of Southern New Mexico, I looked up at the sky and almost fell over in astonishment. Above my head twinkled thousands and thousands of pearly stars arranged in a band diagonally crossing the dark night sky. As a graduate student in the third year of a Ph.D. in astrophysics, I observed the Milky Way for the first time with childlike excitement. It's sad that such a spectacular sight is visible from less than 20 percent of the land in the United States due to the light pollution from our modern cities. Suddenly seeing what the scientific greats of the past had seen led me to understand why the universal fascination with astronomy has been one of the greatest drivers of scientific understanding since the Enlightenment. Yet despite the revolutionary work of Newton, Hubble, and Einstein, we are learning that something is missing in our understanding of the Milky Way – something that today's physicists are explaining with the help of modern scientific computing.

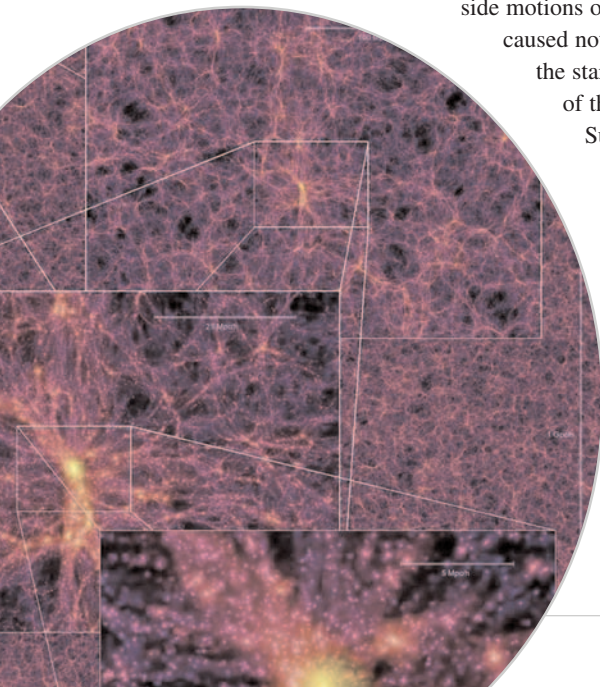
I'm always astonished that astronomers 150 years ago could make precise measurements of positions of stars, since without the push button controls of a modern telescope, I'd be hard-pressed to find Jupiter, let alone stars millions of light-years away. Yet observers can carefully measure the distances to stars using a clever trick, the method of parallax.

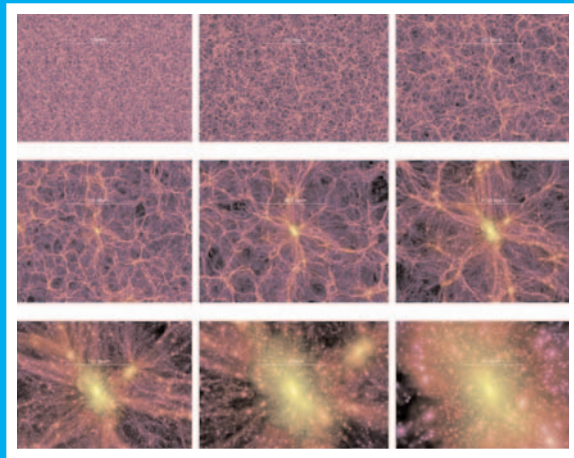
Parallax relies on measuring the side-to-side motions of stars that are caused not by the motion of the stars, but by the motion of the Earth around the Sun. With a little trigonometry, a way of measuring distances using the properties of triangles, the stars reveal their distance from the Earth. Stars are also fairly

simple objects. Simply by measuring their luminosity, a technical term for brightness, we can figure out almost exactly how massive these stars are. Now that we know where all the stars are in the sky and how big they are, we should have a good understanding of our galaxy – or so we thought. Newton predicted in 1687 how objects should move in a gravitational field by showing that the force between two bodies decreases with the square of the distance between them. With modern computers we can take our catalog of all the stars in the galaxy and exactly solve the equations of motion for these bodies. These methods are called N-body simulations. Here computer science and astronomy intersect in a very positive way. The Milky Way has over 200 billion known stars, that is about 40 for every person on Earth. To calculate the interactions between each pair of stars requires massive computational power. Computer scientists have developed a number of techniques to handle this enormous computational challenge. On one hand, two research astronomers, Josh Barnes and Piet Hut, developed a software algorithm, or method of doing a computation, that realizes that we don't have to worry about the forces on stars that are far away from each other since the forces between them are so small. On the other hand, scientists in Japan developed special purpose

"As a graduate student in the third year of a Ph.D. in astrophysics, I observed the Milky Way for the first time with childlike excitement."

Plots from "The Millennium Simulation" by Volker Springel, from Nature 435, 629-636 (02 Jun 2005).





Plots from "The Millennium Simulation" by Volker Springel, from Nature 435, 629-636 (02 Jun 2005).

computer hardware, called GRAPE (GRAvity PipELine) circuits, that are capable of solving only this problem but at the blazingly fast speed of trillions of calculations per second.

When we use these computational tools to simulate the behavior of the Milky Way, we get a very different result than what we can currently observe. The computer tells us that many stars near the edge of our galaxy should be flung off to outer space with enormous velocities, yet we look through a telescope and there they are. For the stars to still be gravitationally bound, as we call objects that are in orbit around the galactic center, there would have to be five times as many stars as we currently can observe. Obviously, something is rotten in the state of Denmark. First, we worry that maybe we've missed some stars. Much like my bookshelf, the universe is filled with cosmic dust that obscures starlight in the visual spectrum. Yet now, with modern telescopes that can see in the infrared (the same part of the spectrum your television remote uses), we can see through the dust and account for those visually obscured stars, just as I was suddenly able to see the Milky Way when I got away from light pollution that obscured my view. Even counting these hidden stars, a large amount of mass is still missing. This dilemma leaves us with the somewhat unpleasant hypothesis that something other than stars that cannot be seen fills our galaxy, and in fact, accounts for most of the mass in our galaxy. We call this missing mass dark matter since it does not emit or reflect any light.

Most astronomers currently believe that dark matter permeates the universe, interacting only with visible matter through the

force of gravity. What is dark matter though? To answer this question physicists have responded with their weapon of last resort...acronyms. The first contender is WIMPs, or Weakly Interacting Massive Particles. These are very heavy particles that litter the universe yet don't really interact with anything.

"This dilemma leaves us with the somewhat unpleasant hypothesis that something other than stars that cannot be seen fills our galaxy, and in fact, accounts for most of the mass in our galaxy."

Despite many attempts to detect particular flavors of WIMPs, no experiment has yet been successful. The second popular candidate is MACHOs, or Massive Compact Halo Objects. These objects could be black holes that live in the halo, the really far out part of the galaxy. Again, none have ever been detected.

With all these acronyms abounding and no proof for anything, scientists are motivated to develop yet more simulations to try to understand what could generate our galaxy and the universe as a whole. By doing ever-fancier simulations they can build alternate models of the universe that would result from a given candidate for dark matter. By comparing these simulations to observations, we are given hints of where to look for the missing matter. The most interesting observation is about the method itself. The modern scientific method is not simply a set of experiments. Astrophysics is now conducted with a rich and interlinked cycle of observations, theory, and computer simulations in an attempt to answer fundamental questions about our universe. In terms of our galaxy, the quest is simple. Do we eventually discover WIMPs, MACHOs, or some other acronym, or do we instead decide that Newton and Einstein were wrong about the laws of gravity? The answer will come down to whether we can find the missing milk in the Milky Way.

**Mala Radhakrishnan –
2005 DOE CSGF
Essay Contest
Honorable
Mention**



The Dream of a Drug-Designing Machine: Prospects, Progress, and Challenges

Picture a scientist at a pharmaceutical company. In your mind she is probably wearing a white lab coat, goggles, and gloves. Perhaps she is holding a pipette or beaker. Yes, there will always be such “experimental” scientists in the field of drug design. But now picture a scientist sitting at a computer, with no lab coat, goggles, or gloves. She has just discovered a new drug – using a computer.

Is this scenario realistic? Can computers design drugs at the click of a button? Not yet. But the hope is that someday, thanks to the rapidly emerging field of structure-based drug design, such methods of drug discovery might become the norm.

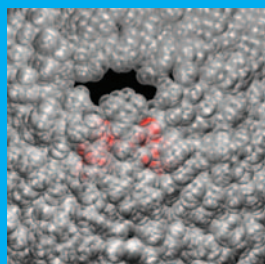
Today, drugs are generally developed by screening countless molecules in test tubes or cells, and seeing if any of them have desired properties. It’s kind of like throwing thousands of darts while blindfolded, and hoping that one of them hits the bulls-eye. Our hope is that eventually, much of this experimentation will be simulated by computers, saving human and chemical resources, time, money, and perhaps some animal testing. Nevertheless, although the field has made considerable progress already, we still have a long way to go.

That being said, how might you design a drug using a computer? As an example, consider HIV, for which computation has already played a role in several FDA-approved drugs. The first step, which generally does not directly involve computation, is to understand how the disease works and how you can interfere with it. In the case of AIDS, the cause is the HIV virus. Like robots-gone-wrong in a horror movie, a virus builds more copies of itself using tools it makes through hijacking your cells. A sensible way to attack a virus, therefore, is to keep these necessary tools from working. One of the tools HIV uses is an enzyme – a little molecular machine – called HIV-protease, which acts like a pair of molecule-sized garden shears that the virus needs to build more copies of itself. HIV-protease is therefore a tool in the virus’ toolbox that we want to inactivate – it is a *good drug target*.

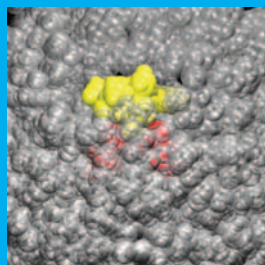
Once a drug target is identified, the next step is to brainstorm how a drug could inactivate it. If you wanted to inactivate a pair of garden shears, you could put a big wad of gum between the blades. As unsophisticated as it may seem, many HIV drugs do just that – they act like a big wad of gum, plugging up HIV-protease so it cannot do its job.

But how do you design a molecule that acts like a wad of gum toward HIV-protease? Like all enzymes, HIV-protease is made up of a unique combination of atoms, which, when bonded to each other, take on a certain shape and have their own footprint of positive and negative charges. As we know, opposites attract; positive and negative charges “stick” to each other. So if a small molecule has certain charges in just the right places and its shape is complementary to the shape of the “blades,” it will be attracted to the protease and will snap into place like a puzzle piece, clogging the molecular shears.

Here’s where computation becomes useful; knowing the shape and the charge pattern of HIV-protease, we can let a computer calculate how strongly a given molecule is attracted to sit between the “blades” of HIV-protease. A computer can screen billions of potential drug molecules




A view of HIV protease. Each atom in the protein is represented by a sphere, and the spheres have been made transparent for clarity. Notice that there is a tunnel through the protease – this tunnel is where other molecules are trapped so that HIV protease can cleave them like a pair of garden shears. The parts of the molecule shown in red are crucial for doing the actual cutting – they are like the “blades” of HIV protease. HIV’s survival depends on HIV protease’s ability to cleave certain molecules.



The same view of HIV protease, although now a computationally designed molecule plugs up the space needed for HIV protease to function properly, and the “blades” – the parts shown in red – are unable to cleave this particular molecule, so it can continue to plug up the tunnel.

HIV protease structure is from pdb code 1kgr: Prabu-Jayabalan M, Nativaika E, Schiffer, CA. Structure. 10(3):369-81, 2002. Image Software used was vmd: Humphrey, W, Dalke, A, Schulten, K. J Molec. Graphics. 14(1):33-38, 1996. Rendering of images was done with Raster3D: Merritt&Bacon. Meth.Enzymol. 227:505-524, 1997. Drug molecule shown is designed by Michael Altman and Bruce Tidor.



many times faster than actual experiments would take, and it may be able to determine which one gives the best fit. Alternatively, the computer may be able to build a potential drug molecule from scratch, working within the rules of chemistry to make a molecule with an ideal shape and charge pattern. Such computational feats make my spine tingle; I smile at the thought of such a drug-designing machine – we could feed a picture of a target into a computer, and out would pop a drug molecule that will definitely bind to and inactivate it. Unfortunately, as of now, we are far from achieving such certainty. Molecules interact with each other in a very complex manner, and we do not fully understand the physics governing these interactions. Bluntly stated, our calculations are just not accurate enough. The simple adage that “opposites attract” does not describe the whole physical story, and defining the “shape” of a molecule is difficult and involves complicated theories such as quantum mechanics. If we were to explicitly include all these interactions, then the calculations would take so long that even the world’s most powerful computers would design drugs far more slowly than the people in lab coats do today! There is a tradeoff between computational efficiency and accuracy. Can we come up with fast but accurate approximations of how molecules interact? Can we design computational algorithms that spit out optimal molecules in a reasonable amount of time? Admittedly,

sometimes my head hurts when I think about the daunting tasks ahead of us on both the scientific and computational fronts.

But there’s more: designing a molecule that binds tightly to a target like HIV-protease is just the beginning. Among other things, this designed molecule must be non-toxic and able to get where it needs to in the body, and it must not bind tightly to other molecules in the body, as this would cause unwanted side effects. These drug properties can also be modeled computationally, adding other dimensions to our problem.

The tasks ahead are formidable, but I am nevertheless impressed at the progress the field has already made; computers are being used right now in the drug-designing process. A starting screening pool of billions of molecules might be computationally narrowed down to a few hundred for experimentalists to try out. Some of these may bind to and inactivate the target. This is a good start, but can we do better? Can we more accurately approximate what makes molecules “stick” to each other like wads of gum? Call me an optimist, but I’m confident that someday, next to the beakers and test tubes in our mental image of a scientist, we’ll all see a computer, with a novel drug molecule peacefully rotating on its screen.

DOE CSGF Essay Contest

	Ian Parrish	Mala Radhakrishnan	Kristine Cochran
--	-------------	--------------------	------------------

Shades of Right

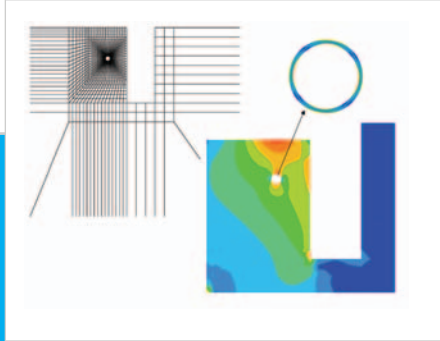
My husband has an amazing knack for always sounding like he has the right answer. Most of the time he does, but sometimes he is unsure, or just plain wrong. I’ve never had that same confidence in my knowledge and as an engineer this deeply disturbs me. It’s not such a big deal in the classroom; sooner or later all answers will be checked against known solutions. In real life, and even in academic research, the significance of being “right” takes on a much deeper meaning. In my field of structural engineering wrong answers can lead to buildings collapsing, dams bursting, and taxpayers spending too much money. Physical experimentation, careful error checking, and past experience traditionally have defined what “right” means in structural engineering.

Today, the use of computers in structural engineering is changing what it means to be “right.” Computers introduce new possibilities for mistakes and demand new means be found to verify our answers.

My first encounter with computational structural analysis was during a course titled “Theory of Structures” that I took as an undergraduate at Johns Hopkins. The course emphasized paper and pencil techniques for analyzing forces and displacements in simple, idealized structures. Midway through the semester I encountered a homework problem for which I could not get my answer to match the one given in the textbook. I toiled for hours and generated nearly a notebook full of calculations but still did not get the solution printed in the back of the book. The next morning, feeling defeated, I went to see the professor. I explained the situation and asked for help. Professor Cox turned on his computer and launched some software he had written. He typed in a few commands and within a couple seconds he informed me that the answer in the book was wrong. I was stunned. Not only did he get an answer in a fraction of the time it took me with a pencil, he was completely confident in it.

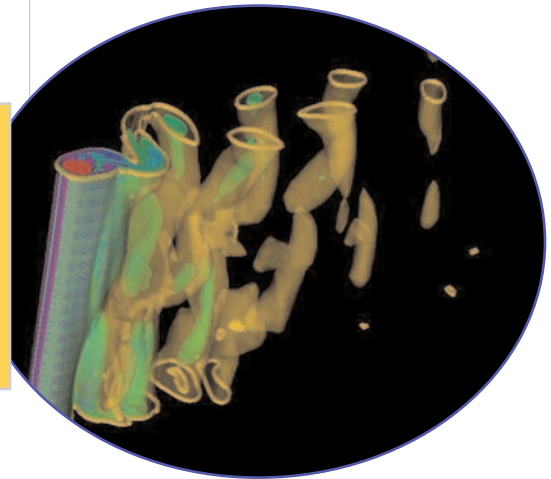
***Kristine Cochran –
2005 DOE CSGF
Essay Contest
Honorable
Mention***





Converting a physical problem into a computer model requires many approximations. In this model of a buried pipe near a trench, the geometry, material behavior and loading have been simplified. The accuracy of the resulting stresses in the soil and pipe is limited by suitability of the approximations.

Though computers allow us to model and visualize complex phenomena, verifying, understanding and interpreting the results often requires a great deal of knowledge and effort. Shown here is a model of the vorticity of fluid flow around a fixed cylinder. Image courtesy of Daniel Turner at the University of Illinois.



His confidence stemmed from the relentless verification he performed on the software during its development. New software must be tested against various problems of known solution and, when possible, against similar

number of equations. Clearly this is not practical, so engineers must pick the most relevant features and model them as well as possible. If we need a very accurate answer, “as well as possible” means one thing. If we need an answer quickly, it may mean something else.

Limiting the physical attributes in a material model is one source of error, but there are other potential sources of error introduced during computational modeling. Imagine that a graph of the chosen material behavior looks like a twisty mountain road, where the horizontal distance represents the deformation and the vertical distance represents the load. The equations that describe such a path are very complex and difficult to solve. Often the exact answer is unobtainable. We must approximate the solution. If we divide the road into segments, each segment might not look so bad. For example, the road between runaway truck ramps may look almost straight, though quite steep. Engineers approximate the solution to material models by chopping the path into segments described by simplified equations. Generally, more complicated approximations or smaller segments lead to more exact answers but longer compute times. At some point though, decreasing the segment size or increasing the complexity stops giving us any more useful information. Engineering sense is essential for obtaining a practical answer.

Some engineers believe that computers diminish the “art” in structural engineering: the engineering sense developed over a long career to discern “right” from “wrong”. In reality, computation allows structural engineers to focus on engineering decisions rather than tedious calculation. It is harder than ever to sort out what it means to be “right” but only because computers have lifted so many limitations. I am sure my husband will continue to think he is “right” most of the time, but I must learn to be content in a world painted in shades of “right.”

software. As the new software passes each test, one gains confidence it will produce correct answers for problems with unknown solutions. Unlike people, computers are not prone to misplaced decimal points or other math mistakes. The types of errors made with computers are often more subtle, but once found and fixed, they will not recur. Troubleshooting may now be done once and for all, rather than for each new problem.

Though the role of the engineer is evolving, computational analysis will never replace a well-trained, intelligent structural engineer. Engineers need to be sure that answers, especially if computer generated, make sense. I first became aware of this when some of my fellow students were working on a highway overpass design project. They had nearly finished their design when the instructor looked over their work. He informed them that their beams and girders were much larger than necessary. After examining their work, the team discovered they had accidentally entered twice the actual load into the analysis program. The instructor’s experience alone was sufficient to bring these faults to light.

In my research, there are seldom “right” answers. Instead, I struggle to provide “right enough” answers. I work with material models – mathematical descriptions of how materials deform when loaded. Before computers, only very simple models were practical. Real materials subjected to hefty loads, large displacements or significant temperature changes react in very complex ways. Full characterization of the material may be possible given an infinite amount of data and an unlimited

**For more information, contact
Martin Edelson at the Krell Institute**

WEB:

www.krellinst.org

PHONE:

515-956-3696

FAX:

515-956-3699

MAIL:

1609 Golden Aspen Drive
Suite 101
Ames, IA 50010

EMAIL:

edelson@krellinst.org

